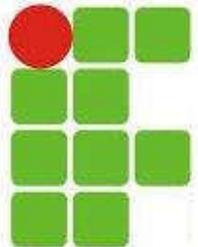

ALGORITMOS

Professor: Diego Oliveira



**Aula 12 -
Laços de Repetição (WHILE)**





WHILE

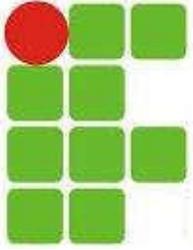
- O laço de repetição WHILE, em português ENQUANTO, é executado enquanto uma condição for verdadeira:

```
public static void main(String[] args) {  
    int i = 0;  
  
    while (i < 10) {  
        System.out.println("i = " + i++);  
    }  
}
```

```
Output - LaçosDeRepetição (run) ×  
run:  
i = 0  
i = 1  
i = 2  
i = 3  
i = 4  
i = 5  
i = 6  
i = 7  
i = 8  
i = 9
```



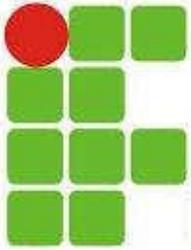
- Observe que o 10 não é impresso!



WHILE

- Um erro muito comum quando se utiliza o WHILE é não atualizar a variável que está sendo verificada, o que causa um laço infinito e o programa ‘trava’.
- Por isso o ‘i’ no código anterior está sendo incrementado de um em um DENTRO DO LAÇO!





WHILE

- O WHILE também pode ser utilizado para executar laços dos quais não sabemos o número de repetições necessárias, como por exemplo um

MENU:

```
public static void main(String[] args) {
    Scanner leia = new Scanner(System.in);
    int op=0;

    while (op != 3) {
        System.out.println("1-START GAME");
        System.out.println("2-LOAD GAME");
        System.out.println("3-EXIT GAME");
        op = leia.nextInt();
    }
}
```





WHILE

- Saída do slide anterior para entradas '1', '2' e '3':

```
Output - LaçosDeRepetição (run) X
RUN:
1-START GAME
2-LOAD GAME
3-EXIT GAME
1
1-START GAME
2-LOAD GAME
3-EXIT GAME
2
1-START GAME
2-LOAD GAME
3-EXIT GAME
3
BUILD SUCCESSFUL (total time: 4 seconds)
```





WHILE

- Um outro exemplo do uso do WHILE seria para calcular a média de uma turma, da qual não sabemos ainda o número de alunos:

```
public static void main(String[] args) {  
    Scanner leia = new Scanner(System.in);  
    int numeroAlunos = 0;  
    double mediaTurma = 0;  
    double mediaAluno = 0;  
    while (mediaAluno != -1) {  
        System.out.println("DIGITE A MÉDIA DO ALUNO: ");  
        mediaAluno = leia.nextDouble();  
        if (mediaAluno != -1) {  
            numeroAlunos++;  
            mediaTurma += mediaAluno;  
        }  
    }  
    mediaTurma = mediaTurma / numeroAlunos;  
    System.out.println("MÉDIA DA TURMA = " + mediaTurma);  
}
```





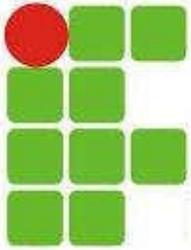
WHILE

- Saída do código anterior:

```
Output - LaçosDeRepetição (run) ×
run:
DIGITE A MÉDIA DO ALUNO:
5
DIGITE A MÉDIA DO ALUNO:
5
DIGITE A MÉDIA DO ALUNO:
6
DIGITE A MÉDIA DO ALUNO:
6
DIGITE A MÉDIA DO ALUNO:
-1
MÉDIA DA TURMA = 5.5
BUILD SUCCESSFUL (total time: 5 seconds)
```



- Observe que foi definido para sair do laço caso a média digitada fosse -1, pois não é uma nota válida



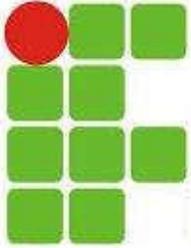
Números Aleatórios

- Em Jogos se utiliza muito a geração aleatória de números para criar objetos em posições aleatórias no cenário ou, mais recentemente, gerar cartas em LOOT
- Em Java podemos utilizar o seguinte código:

```
Random ale = new Random();  
int aleatorio = 1+ale.nextInt(100);
```

- Este código gerará um número aleatório entre 1 e 100

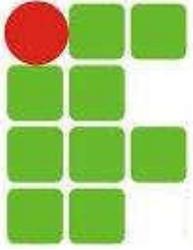




Exercício

- Faça um jogo no qual será gerado um número aleatório entre 1 e 10000.
- O usuário poderá chutar até 20 vezes
- Caso ele chute um número menor que o gerado aleatoriamente, imprima “é um número MAIOR”, se ele digitar um maior imprima “é um número MENOR”.
- Quando ele acertar, uma mensagem de PARABÉNS deve ser impressa
- Caso ele não acerte em 20 tentativas imprima “Infelizmente você perdeu!”





Perguntas?

